

# Recommendations API

The Monetate Recommendations Rest API bring recommendations from your product catalog to any digitally connected channel. This means that you can send a request to the API with a feed ID and a product ID in order to return a set of recommended products.

## Response Codes

When you make a request to the Recommendations API, Monetate returns a response code. This code can vary based on your request. Use the following definitions to help you diagnose failures, invalid requests, errors, and successes:

- **200** (Success): Returns a list of recommendations
- **400** (Bad Request): Your request included a parameter that is unsupported.
- **500** (Error): An unknown error has occurred. Please try again or contact your account manager for more information.

## Product Recommendations API Endpoint

The Product Recommendations Endpoint allows you to send information to the API (such as feed ID and product ID) and return recommendations. An example request and information about the required parameters are available below.

### Request

The request format is as follows:

```
https://reco.monetate.net/catalog/2/<account_id>/<instance>/<domain>/recos?feed_id=<feed_id>
```

Here's an example URL:

```
https://reco.monetate.net/catalog/2/a-524342-ee/p/example.com/recos?feed_id=0329834&pid=23
```

### Required Parameters

The required parameters are as follows:

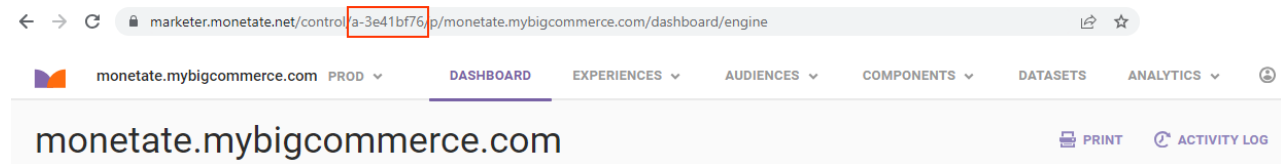
- `<account_id>` – A unique ID included in the application URL or provided to you by the Support team
- `<instance>` – Instance type, which can only be *p* for production or *d* for development
- `<feed_id>` – Merchandiser set ID
- `<domain>` – The domain for your account
- `<product_id>` – A product ID from your Monetate product catalog, which you can find searching your product catalog



A product ID (PID) is not required for every merchandiser set type (algorithm), but it is required for some. Viewed and Later Purchased, Viewed and Also Viewed, and Purchased and Also Purchased types all require a PID. Nonbehavioral merchandiser set types such as Newest Products do not require a PID. In this instance, including one does not change the results that Kibo Personalization returns.

## Finding the Account ID

You can find your account ID by viewing your account URL. The image below demonstrates where to find it.



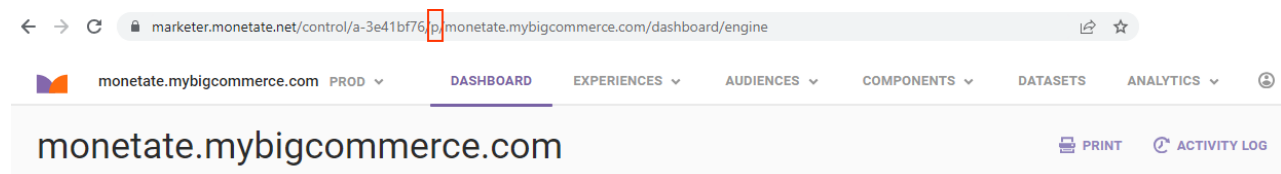
Copy and paste this value into the API call.

## Finding the Instance

Log in to your Monetate account. In the browser address bar, search for the letter *p* or the letter *d* following your account ID.

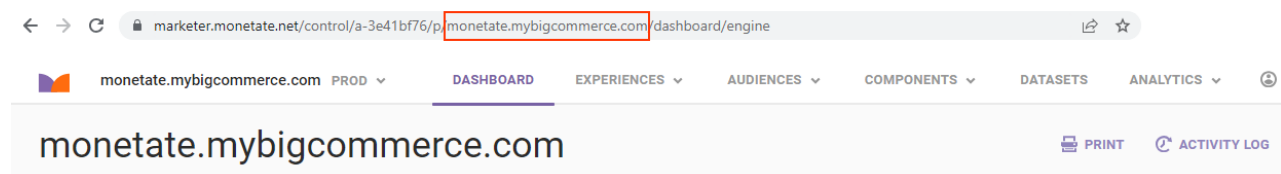
- *P* for production
- *D* for development

Use this to determine the value that you include in `<instance>`.



## Finding the Domain

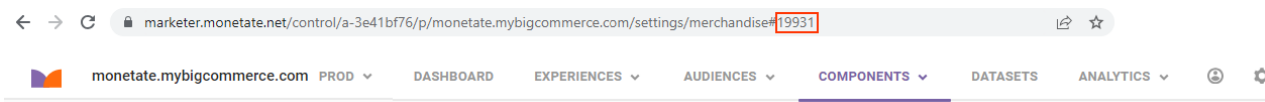
Log in to your Monetate account. In the browser address bar, search for the domain contained within the forward slashes after the instance.



## Finding the Feed ID

To find the feed ID, navigate to **COMPONENTS > Product Catalog (Old Spec)** and then click the **Merchandiser**

**Sets** tab. Next, click the merchandiser set to view its Merchandiser Set detail page, and then examine the feed ID in the URL bar of your browser. Use the ID after the # as the feed ID.



## Response

The JSON code sample below depicts a typical response that you can expect if you have sent a successfully authenticated request to the Recommendations API.

```

{
  data: {
    name: "My Merchandiser Set",
    feed_type: "bestsellers",
    fallbackfeedtype: "newest",
    fallbackfeedid: 8940,
    apiversion: "2",
    productsdata: [
      {
        "producturl": "https://www.example.com/shop/mens-ultra-fastpack-ii-mid-gore-tex-cdl8",
        "rank": "5",
        "alternateprice": "",
        "name": "MEN'S ULTRA FASTPACK II MID GORE-TEX;",
        "price": "155.00",
        "outofstock": false,
        "pid": "CDL8",
        "securethumbnailimage": "https://images.example.com/is/image/example/CDL8GQWhero?370x370",
        "cartable": true,
        "thumbnailimage": "https://images.example.com/is/image/example/CDL8GQWhero?371x371"
      },
      {
        "producturl": "https://www.example.com/shop/mens-hedgehog-fastpack-mid-gore-tex-cxu5",
        "rank": "8",
        "alternateprice": "",
        "name": "MEN'S HEDGEHOG FASTPACK MID GORE-TEX;",
        "price": "130.00",
        "outofstock": false,
        "pid": "CXU5",
        "securethumbnailimage": "https://images.example.com/is/image/example/CXU5DZAhero?370x370",
        "cartable": true
      }
    ],
    noitemsaction: "use-fallback-feed",
    notenoughitemsaction: "use-fallback-feed"
  },
  ref: ""
}

```

## Response Fields

This table contains the response fields and a description of each.

Field	Description
Product URL	URL of the product detail page for a returned product.

Field	Description
Rank	Rank of the product in sequence. The algorithm ranks products based on the strength of their association. You can use these for sorting the products when you place them into a template.
Price	Full list price for the product. The API returns the currency sent in the original catalog load.
Alt-Price	Sale price indicator.
Name	Name of the returned product.
Out of Stock	Boolean indicator (true/false) of whether or not the returned product is in stock.
Cartable	Boolean indicator (true/false) of whether or not the returned product can be added to a cart.
PID	Product ID for the returned product.



`fallback_feedback_type` is useful when you need to deliver a fallback set for recommendations on the Web. That means, if zero products are returned for a merchandiser set, you can use a fallback set to make sure products are always displayed within an endcap. For an example of this, refer to the snippet above. You can ignore this for omnichannel solutions.